Analiza si vizualizarea datelor

Nicoleta ROGOVSCHI

nicoleta.rogovschi@parisdescartes.fr

Isometric feature mapping (Isomap)

Outline

- Introduction and definitions
- Algorithm
- Example
- Conclusions

Dimension reduction via feature extraction

Two main types of methods :

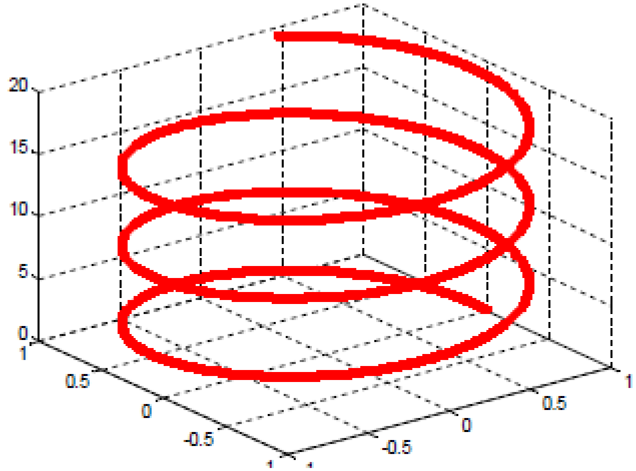
- **Linear Methods**
  - Principal Components Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Multi-Dimensional Scaling (MDS)
  - …
- **Non-Linear Methods**
  - → Isometric feature mapping (Isomap)
  - Locally Linear Embedding (LLE)
  - Kernel PCA
  - Spectral clustering
  - Supervised methods (S-Isomap)
  - …

Introduction

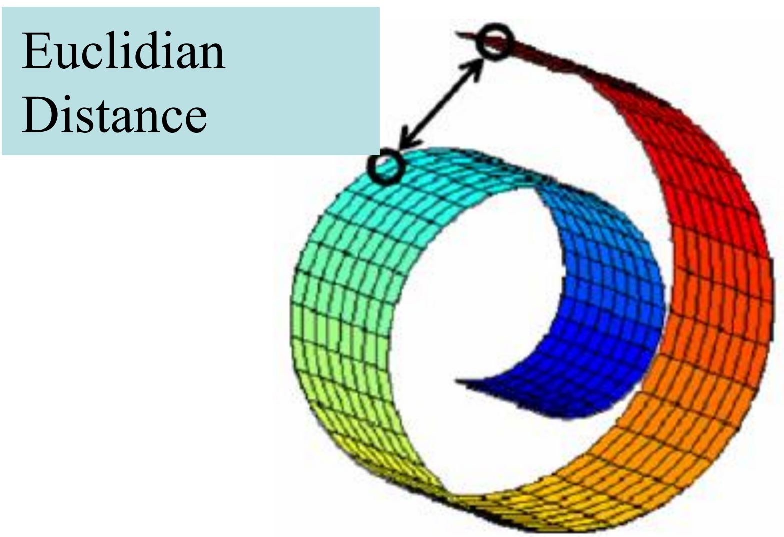
- Techniques as LDA, PCA and their variants perform a global transformation of the data (rotation/translation/rescaling)
  - These techniques assume that most of the information in the data is contained in a linear subspace.
  - Which approache to use when the data is actually embedded in a non-linear subspace (a low-dimensional manifold).

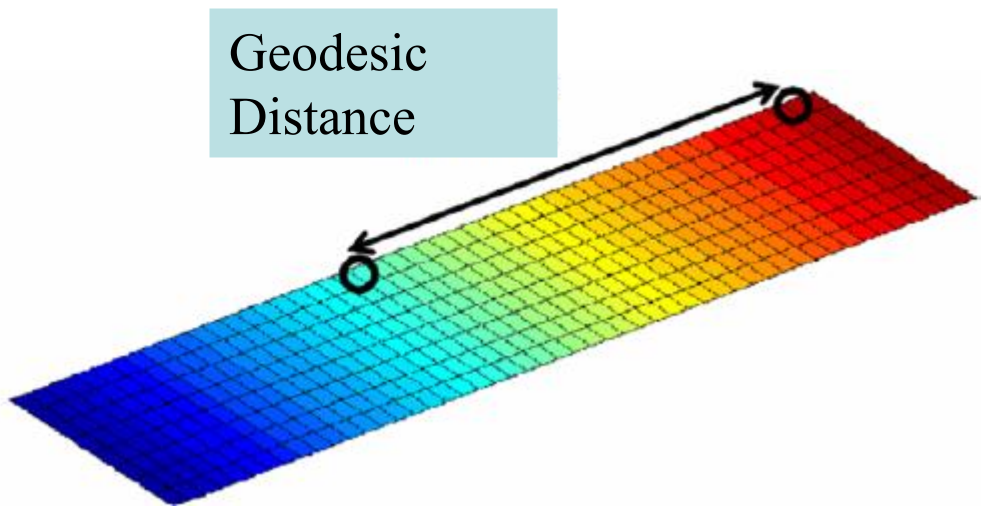
Introduction

- The PCA can not discover the structure of a data set in a spiral form



Euclidian Distance vs. Geodesic Distance





Introduction

- The goal of Isomap is to find a non-linear manifold containing the data
- We use the fact that for close points, the Euclidean distance is a good approximation to the geodesic distance on the manifold
- We build a graph connecting each point to its *k* nearest neighbors

Introduction

- The lengths of the geodesics are then estimated by searching the length of the shortest path between two points in the graph
- Thereafter we apply MDS to obtained distances to determine a position of points in a space of reduced dimensions

ISOMAP

- ISOMAP [Tenebaum et al. 2000]
  - For neighboring samples, Euclidian distance provides a good approximation to geodesic distance
  - For distant points, geodesic distance can be approximated with a sequence of steps between clusters of neighboring points

ISOMAP

ISOMAP operates in three steps:

1. Build the neighborhood graph G
2. For each pair of points of G, compute the shortest path (the geodesic distance)
3. Using the classical MDS on geodesic distances

Euclidian Distance -> Geodesic Distance

ISOMAP Algorithm

- Step 1
  - Build the neighborhood graph based on the distances $d_X(i,j)$ in the input space X.
  - This can be performed in two different ways:
    - Connect each point to all points within a fixed radius ε
    - Connect each point to all of its k nearest neighbors
  - A weighted neighborhood graph G is obtained, where $d_X(i,j)$ is the weight of each edge between neighboring points

ISOMAP Algorithm

- Step 2
  - Estimate the geodesic distances $d_M(i,j)$ between all pair of points on the manifold M by computing their shortest path distances $d_G(i,j)$ in the graph G.
  - It can be done using Dijkstra's algorithm or the Floyd algorithm.

ISOMAP Algorithm

- Step 3

  - Apply classical MDS to the matrix of graph distances D.

Complexity of ISOMAP

- For large datasets ISOMAP can be quite slow :

- Step 1: Complexity of k-nearest neighbors $O(n^2 D)$

- Step 2 : Complexity of the Djikstra algorithm $O(n^2 logn + n^2 k)$

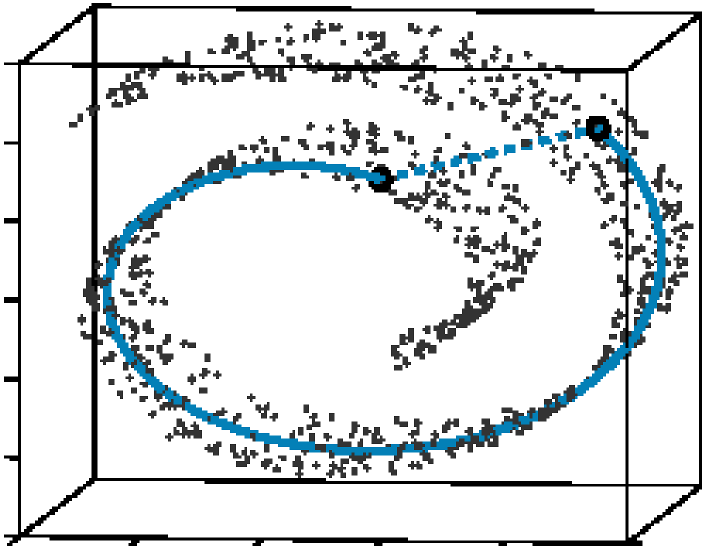
- Step 3 : MDS complexity $O(n^2 d)$

The «Swiss roll» dataset

- The «Swiss roll» data set contains 20000 points.

- In this figure we present a sample of 1000 points.

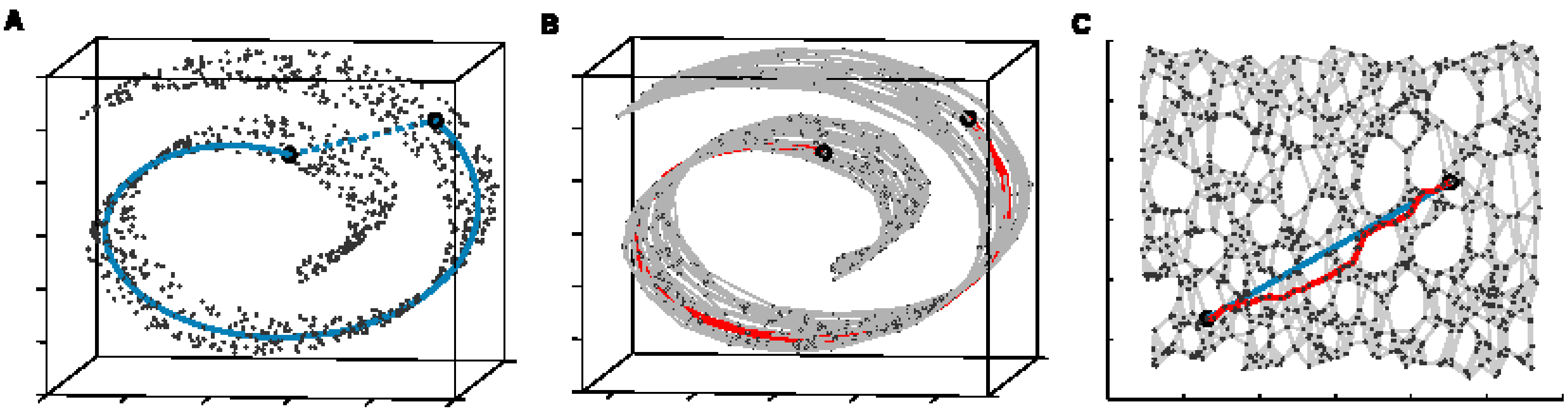
- Thereafter we will represent on this example the development of the Isomap algorithm.



Construction of the neighborhood graph G

K- nearest neighbors (K=7)

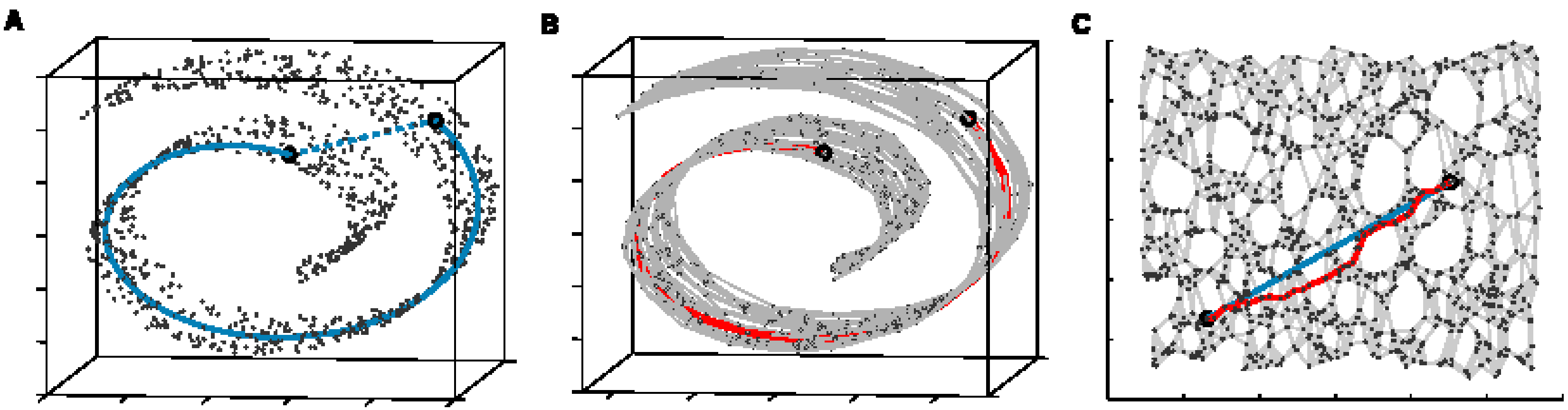
D_G is a matrix of Euclidean distance 1000 x 1000 of two neighboring points (Figure A)



[Tenebaum et al. 2000]

Calculation of shortest paths in G

D_G is a matrix of geodesic distances of two arbitrary points along the manifold M (Figure B)

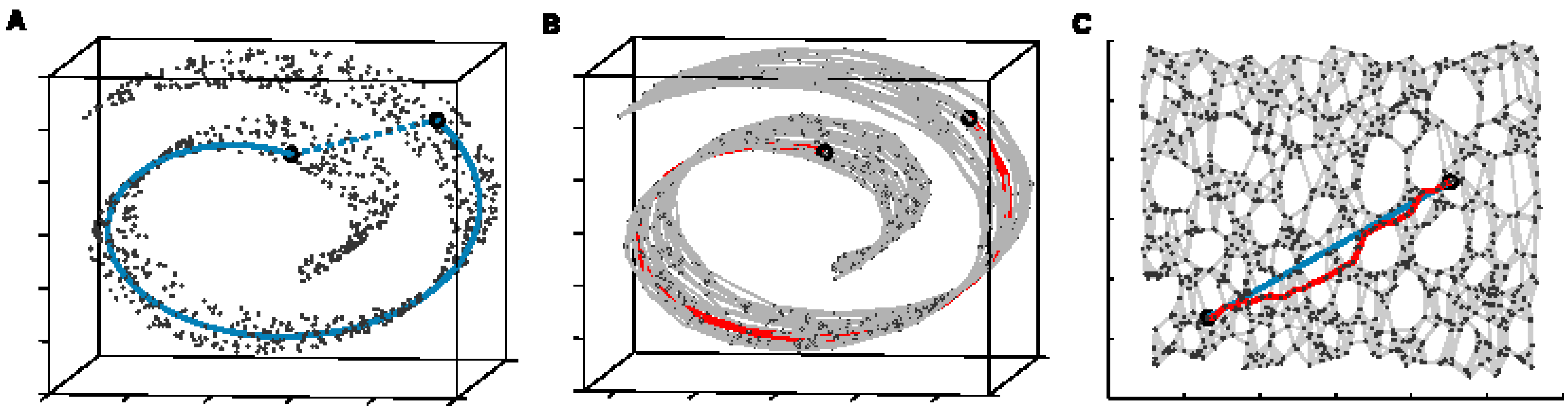


[Tenebaum et al. 2000]

Using MDS to represent the graph in R^d

Find a Euclidean d-dimensional space Y

that preserves the pairwise distances (Figure C)

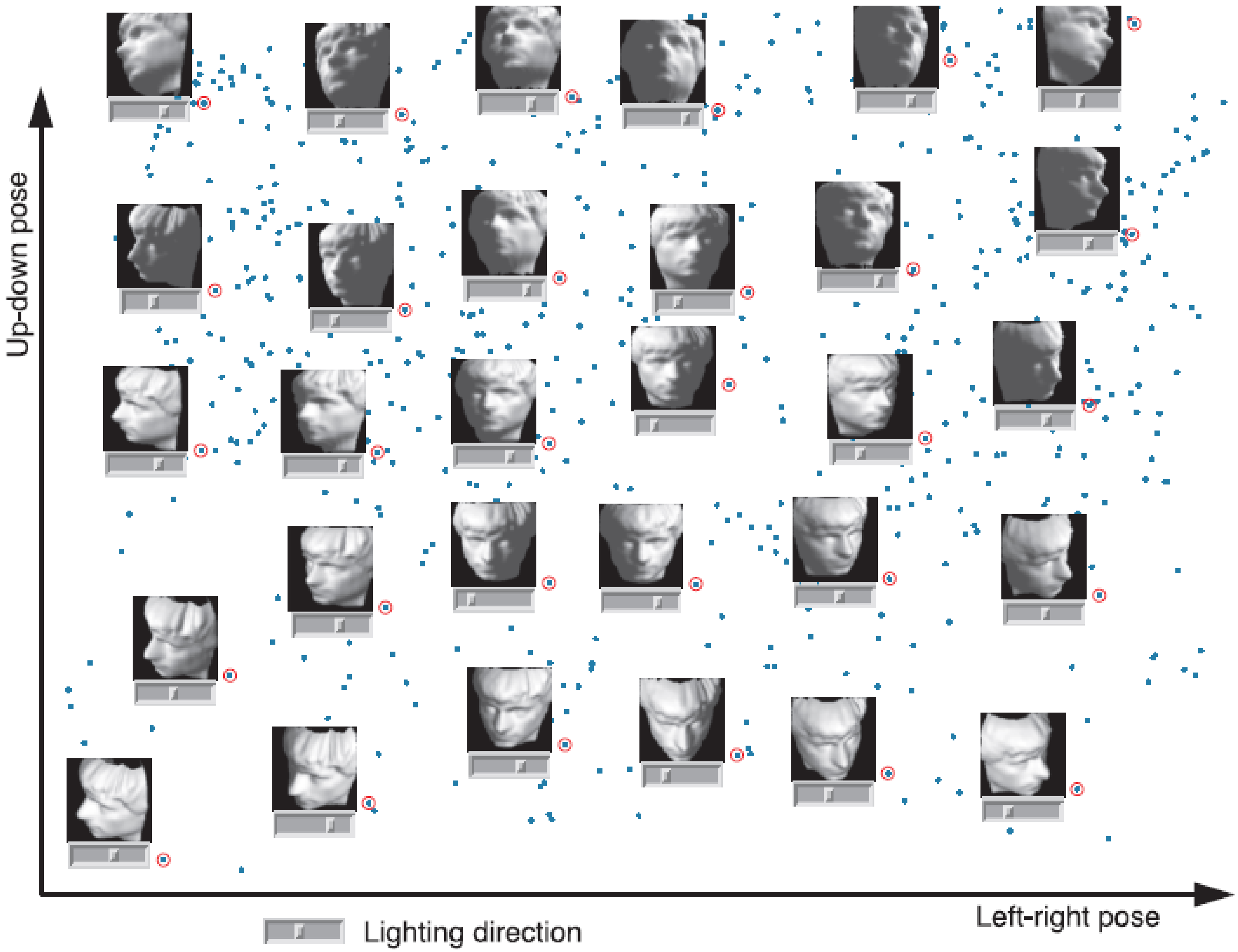


[Tenebaum et al. 2000]

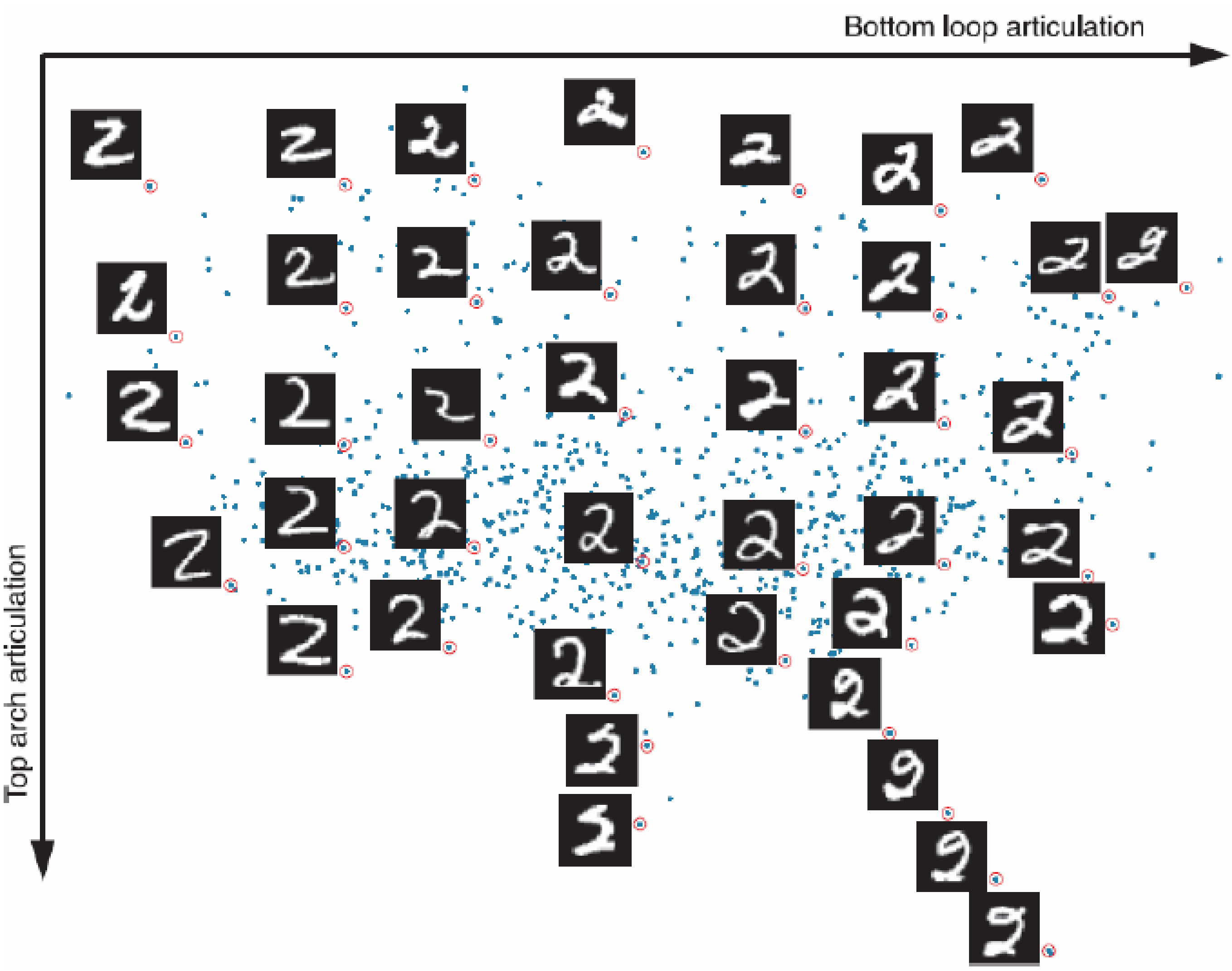
Example on images

For each image we have 64x64 = 4096 pixels

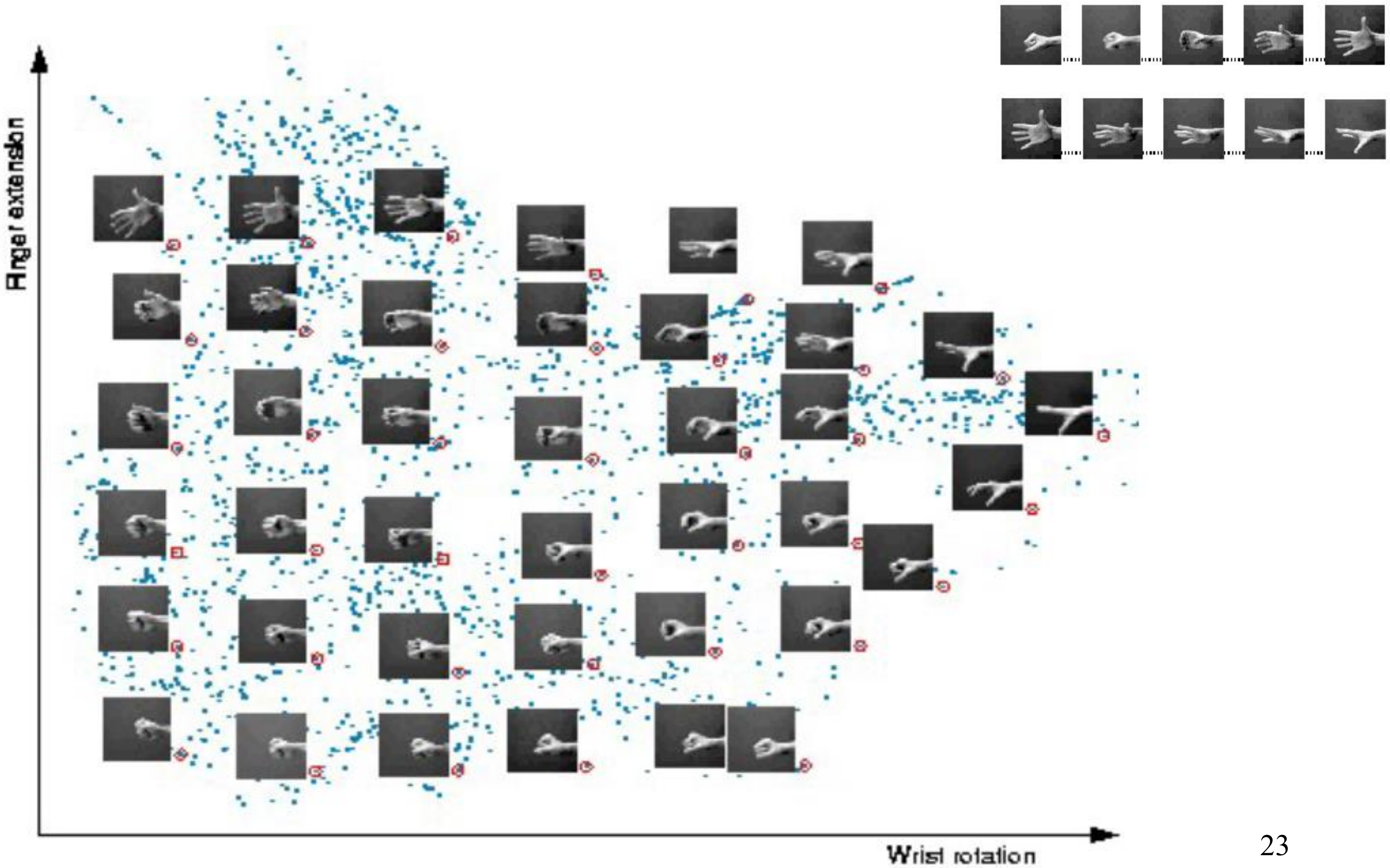
Results



Results



Resultats



Conclusions

- Avantages
  - Non-linear
  - Non-iterative
  - Preserves the global properties of the data
- Limitations
  - Sensitive to noise
  - Parameters to set: k or ε
  - Relatively slow for large data sets
  - k must be high to avoid " linear shortcuts " near regions of high curvature of the surface

Locally Linear Embedding (LLE)

Dimension reduction via feature extraction

Two main types of methods :

- **Linear Methods**
  - Principal Components Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Multi-Dimensional Scaling (MDS)
  - …
- **Non-Linear Methods**
  - Isometric feature mapping (Isomap)
  - Locally Linear Embedding (LLE)
  - Kernel PCA
  - Spectral clustering
  - Supervised methods (S-Isomap)
  - …

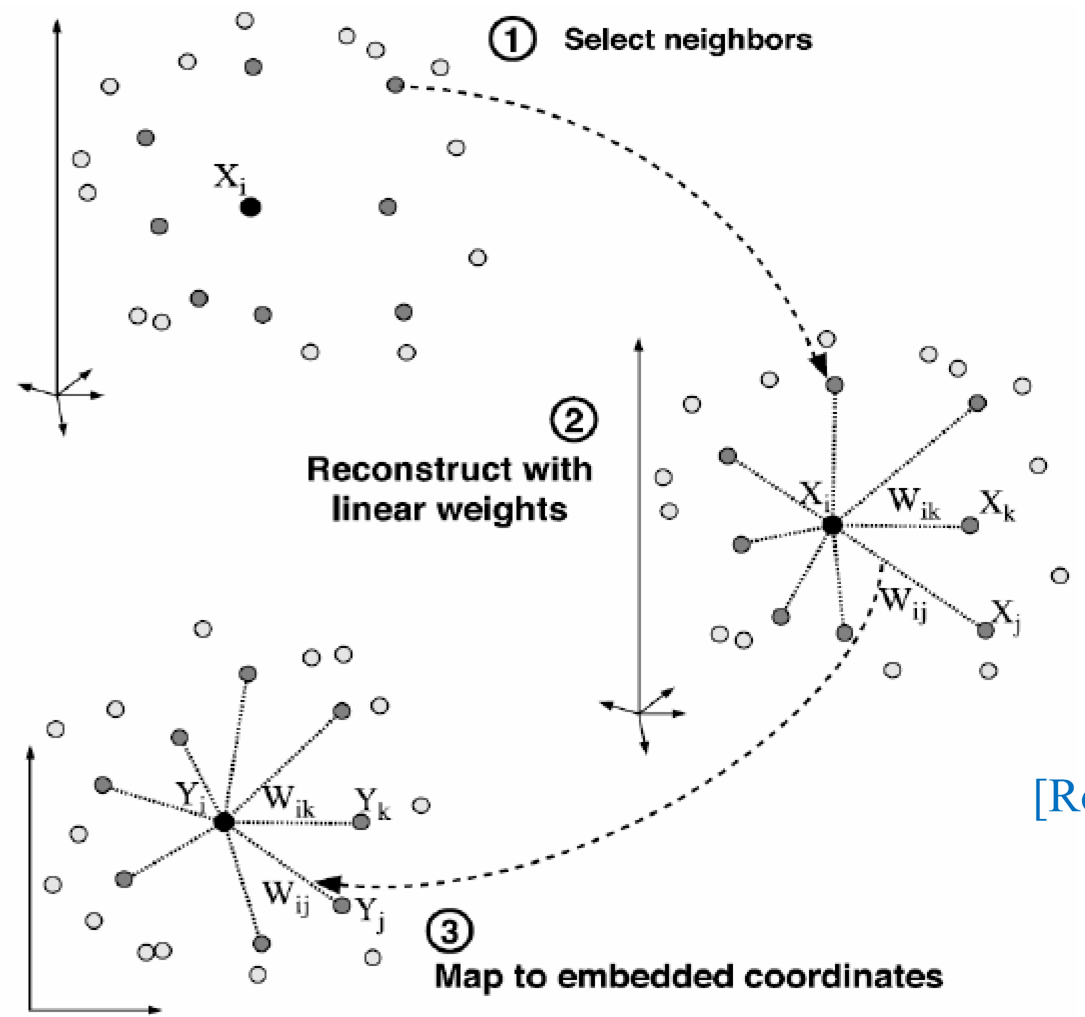
Locally Linear Embedding (LLE)

- LLE ("Locally Linear Embedding") addresses the same problem as Isomap by a different way.

- LLE preserves the local properties of the data representing each point by a linear combination of its nearest neighbors.

- LLE builds a projection to a linear low dimensional space preserving the neighborhood.

LLE Algorithm



[Roweis and Saul 2000]

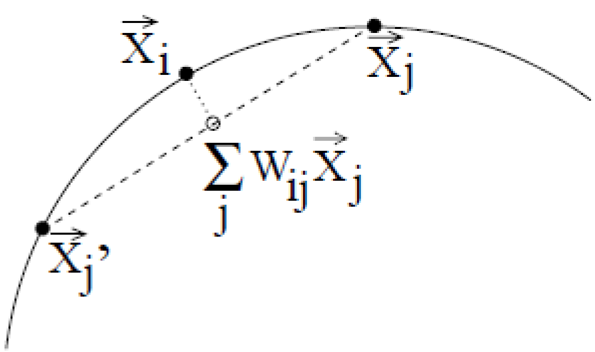
LLE Algorithm

LLE operates in 3 steps:

- Computes the k nearest neighbors

- Computes the weights needed to reconstruct each point using a linear combination of its neighbors

- Projects the results using the new found coordinates.



LLE Algorithm

- The local geometry is modeled by linear weights that reconstruct each data point as a linear combination of its neighbors
- Reconstruction errors are measured with this cost function

$$\varepsilon(W) = \sum_{i=1}^{N} \left| X_i - \sum_j W_{ij} X_j \right|^2$$

  - where weights W_{ij} measure the contribution of the j-th example to the reconstruction of the i-th example

- Weights W_{ij} are minimized subject to two constraints :
  1) Each data point is reconstructed only from its neighbors
  2) Rows of the weight matrix sum up to one : $\sum_j W_{ij} = 1$

LLE Algorithm

- We seek to find *d*-dimensional coordinates Y_i that minimize the following cost function:

$$\phi(Y) = \sum_{i=1}^{N} \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$

Parameter Estimation

- Consider a particular sample x with k nearest neighbors η_j and reconstruction weights w_j (that sum up to one). These weights can be found in three steps :
  - Step 1 : Compute the neighborhood correlation matrix C_{jk} and its inverse C^{-1}

$$C_{jk} = \eta_j^T \eta_k$$

  - Step 2 : Compute the Langrage multiplier λ that enforces the constraint $\sum_j w_j = 1$

$$\lambda = \frac{1 - \sum_{jk} C^{-1}{}_{jk} (x^T \eta_k)}{\sum_{jk} C^{-1}{}_{jk}}$$

  - Step 3 : Compute the reconstruction weights as:

$$w_j = \sum_k C^{-1}{}_{jk} (x^T \eta_k + \lambda)$$

Parameter Estimation

- The embedding vectors Y_i are found by minimizing the cost function:

$$\phi(Y) = \sum_{i=1}^{N} \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$

- To make the optimization problem well-posed we introduce 2 constrains:

$$\sum_j Y_j = 0 \qquad\qquad \frac{1}{N}\sum_i Y_i Y_i^T = I$$

- Which allows to expresse the cost function as:

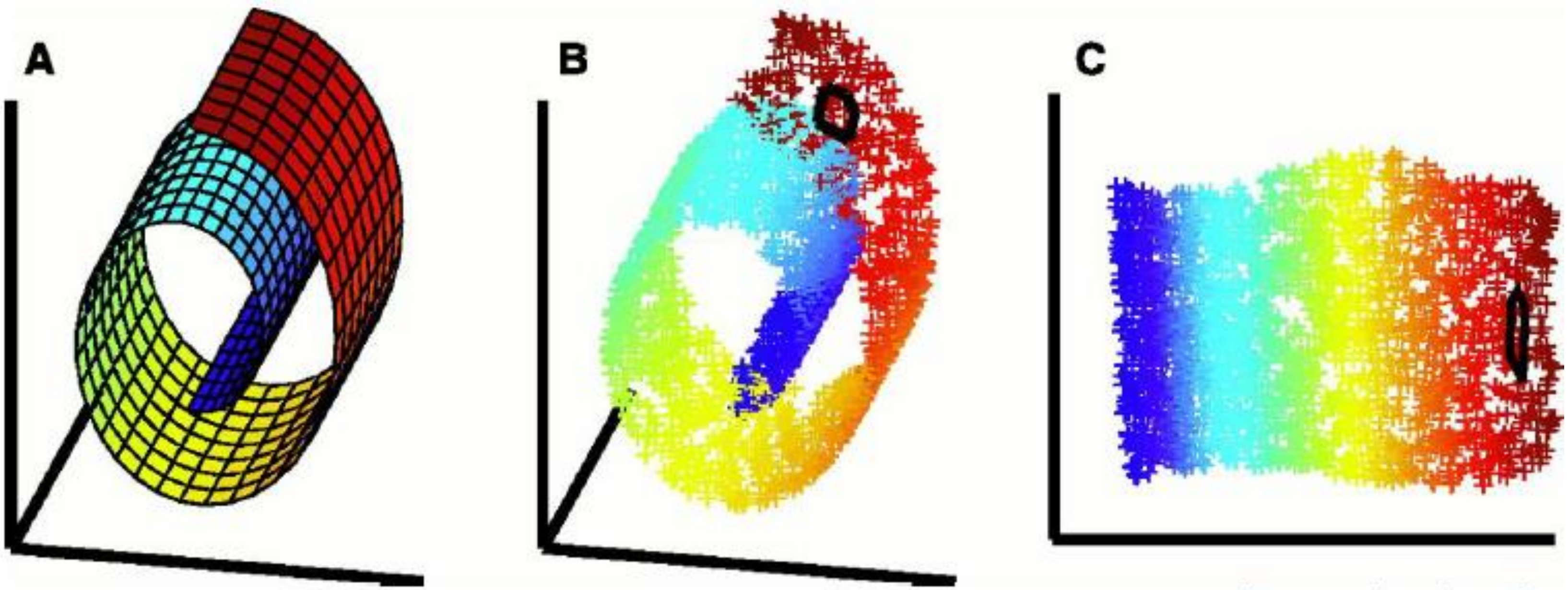
$$\phi(Y) = \sum_{ij} M_{ij} (Y_i^T Y_j)$$

where $M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}$

- δ_{ij} is equal to 1 if i=j and equal to 0 otherwise.

- The optimal embedding is found by computig the bottom *d+1* eigenvectors of matrix *M.*

Examples on LLE



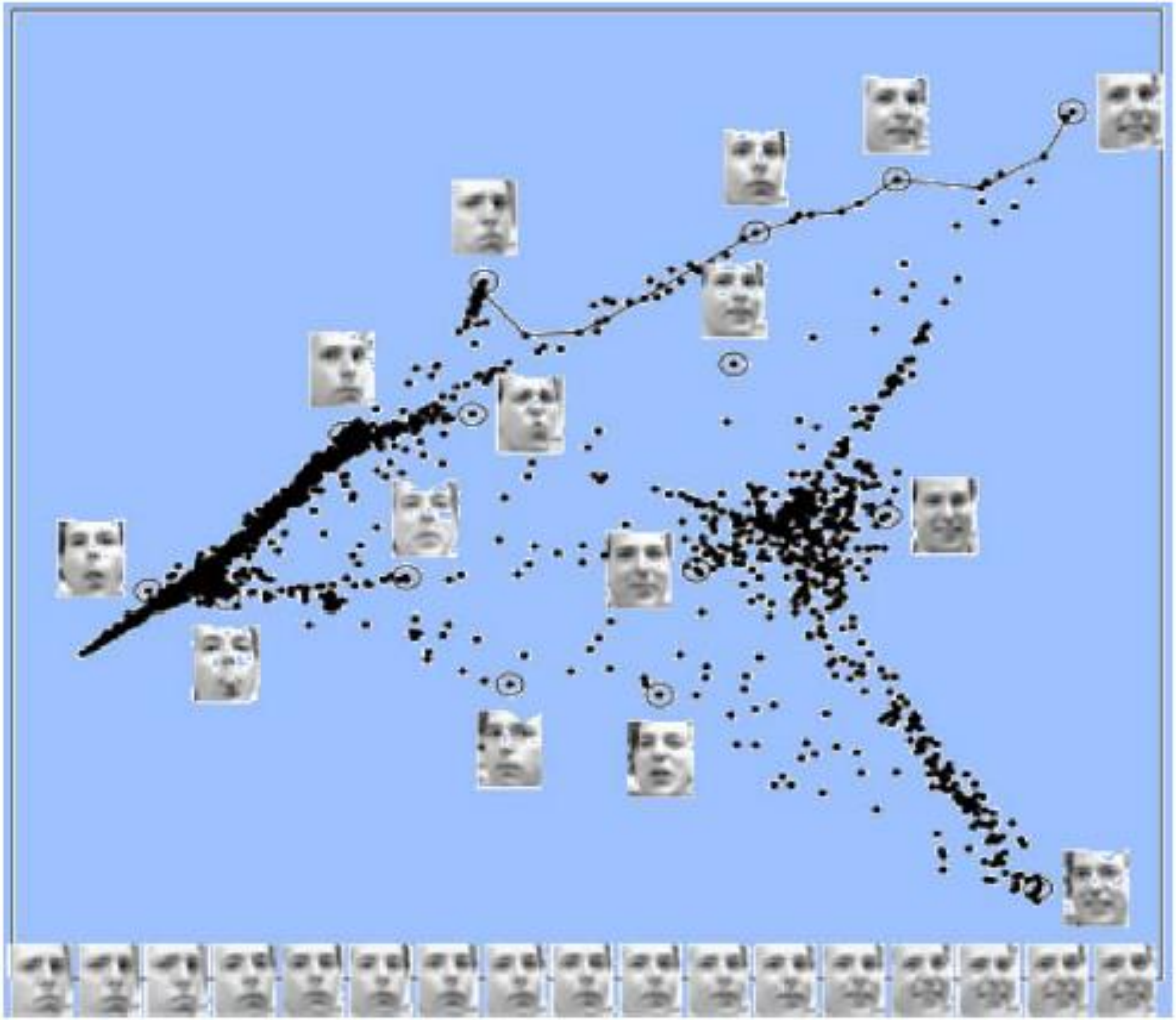
[Roweis and Saul, 2000]

Results

- Initial points represent images of faces.

- In the 2-dimensional space, these images are grouped according to the position, lighting and expression.

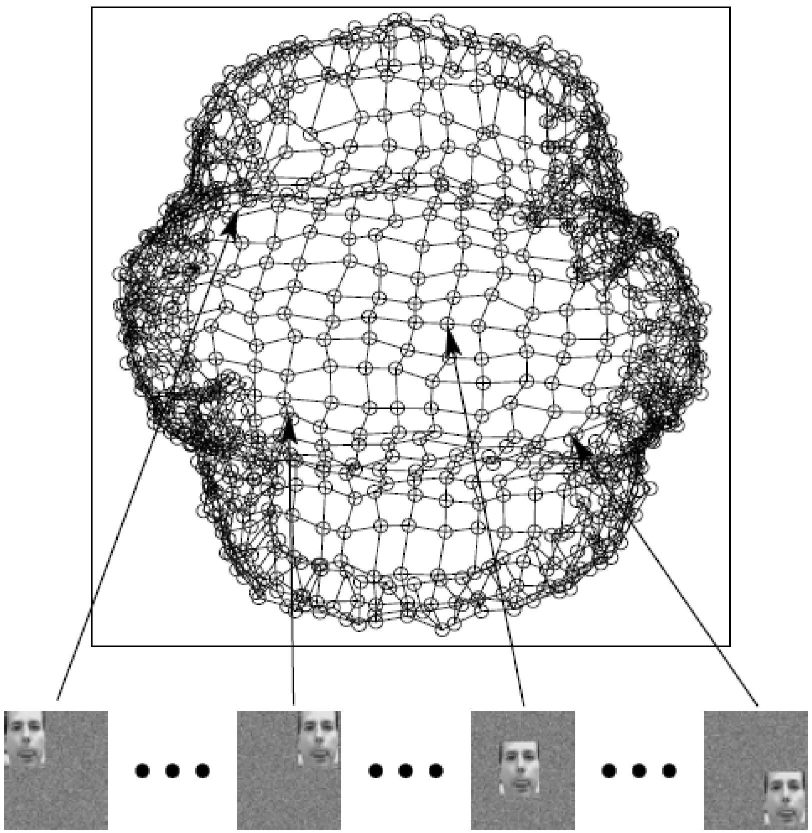
- Images placed in the bottom of the figure correspond to successive points encountered on the line at the top right, sweeping a continuum of facial expression.

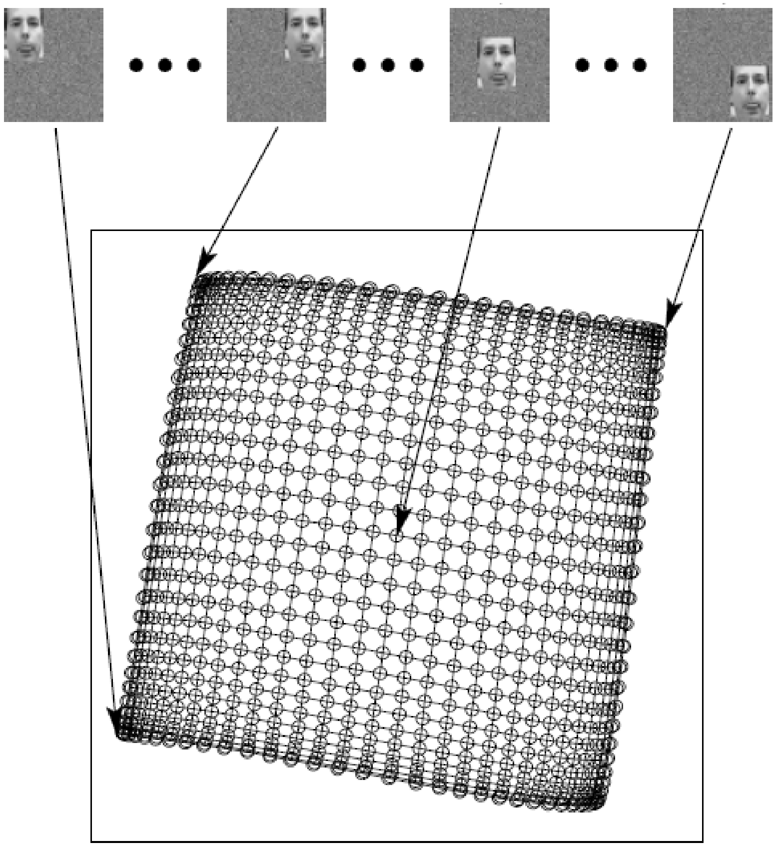


[Roweis and Saul 2000]

Results

- PCA vs LLE



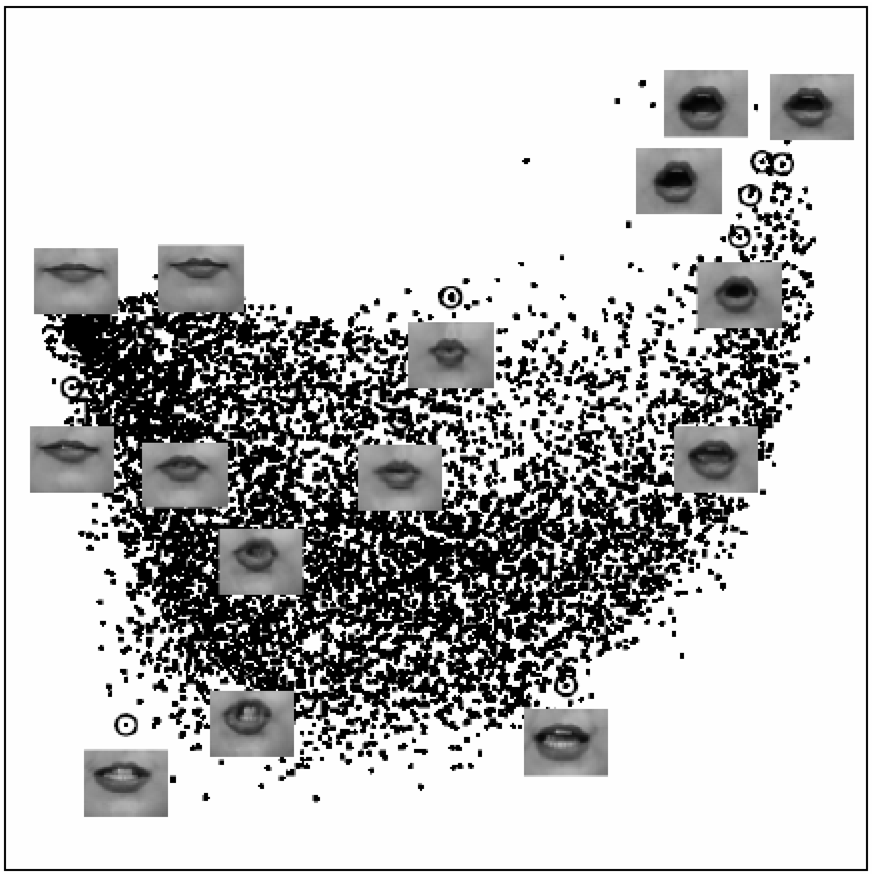


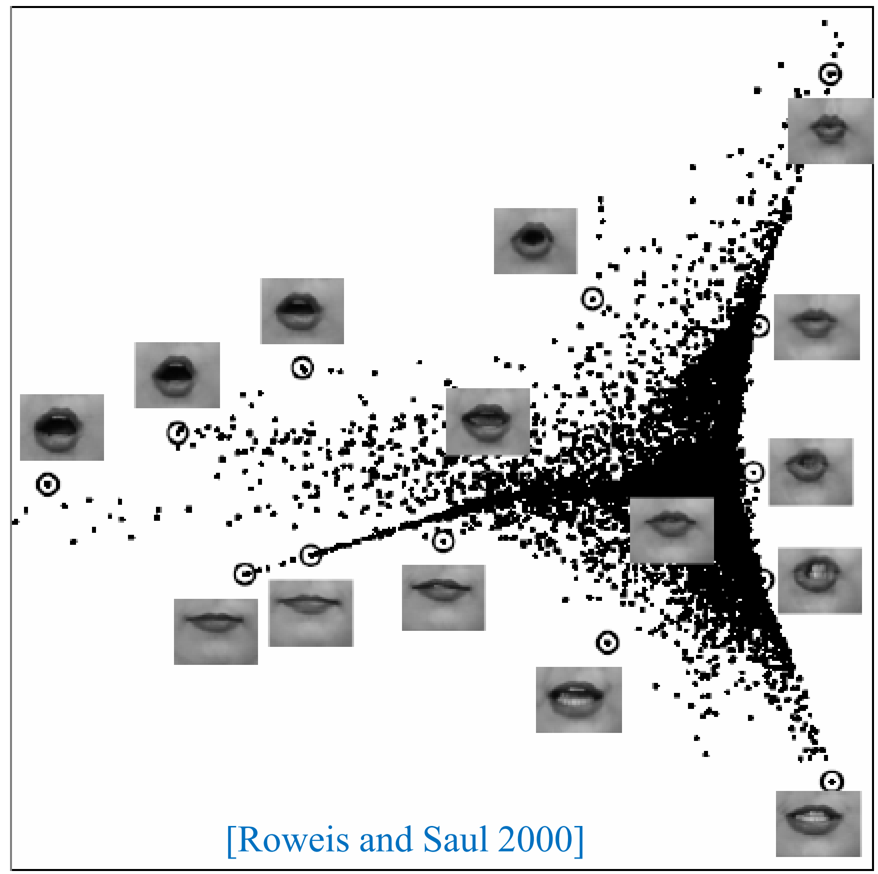
[Roweis and Saul 2000]

Unlike PCA, LLE preserves local topology!

Results

- PCA vs LLE





Conclusions

- LLE is a nonlinear technique that preserves the local properties of the data representing each point by a linear combination of its nearest neighbors in a new space of reduced dimensions.

- LLE operates in 3 steps:
  - Computes the *k* nearest neighbors
  - Computes the weights needed to reconstruct each point using a linear combination of its neighbors
  - Projects results using the new found coordinates

Conclusions

- Sensitive to noise
- Parameters to set: k
- Relatively slow for large data sets

References

- J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319-2323, 2000.
- Sam Roweis & Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323-2326, 2000.

Rappel MDS

We have two types of MDS techniques :

- Metric MDS (classical MDS)
  - We assume that D is the matrix of square distances.
- Non-metric MDS
  - Deals with more general measures of dissimilarity.

ISOMAP Algorithm

- Step 3
  - Apply classical MDS to the matrix of graph distances D.
  - The coordinate vectors y_i are choosen to minimize the following cost function:

$$E = \left\| \tau(D_G) - \tau(D_Y) \right\|_{L^2}$$

    - Where D_Y denotes the matrix of euclidian distances

$\{d_y(i,j)=\| y_i - y_j \|\}$ and operator τ converts distances to inner products:

$$\tau = -HSH/2$$

    - where S is the matrix of squared distances au carré $\{S_{ij}=D^2_{ij}\}$ and H is a centering matrix, defined as :

$$H = I - \frac{1}{N} ee^T; \qquad e = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \end{bmatrix}^T$$

  - The global minimum of E is obtained by setting the coordinates y_i to the top *d* eigenvectors of the matrix $\tau(D_G)$.